

REMARKS

Applicant is in receipt of the Office Action mailed April 26, 2005. Claims 1 – 69 were rejected. Claims 1 – 69 remain pending in the application. Reconsideration of the present case is earnestly requested in light of the following remarks.

Telephone Interview Summary

On Tuesday, June 28, Mark S. Williams, Patent Agent Reg. No. 50,658, conducted a telephone interview with Examiner Huynh. Applicant noted that the client, National Instruments Corporation, is a pioneer in graphical programming, that the both the present application and the cited application (Washington) were prepared by the Agent's firm, and that Agent is very familiar with the subject matter of Washington, which is particularly directed to the automatic generation of graphical programs. Applicant clarified the meaning of such terms as "graphical program" and "graphical program node", noting that as defined, a graphical program comprises "a plurality of interconnected nodes that visually represent functionality of the graphical program", and that a single graphical program node is *not* considered a graphical program. Applicant also emphasized that prior to the present invention, each graphical program had its own front panel (GUI), that Applicant is not aware of any other graphical programming product or system that allows two concurrently executing graphical programs to send their respective program outputs to a single graphical user interface, that the invention was not that easy to implement, and that National Instruments considers the present invention to be quite novel and non-obvious, and of great utility. The Examiner agreed that these points were germane to the present case and cited art, and suggested that Applicant send an official Response After Final including these arguments and issues, and that indicated that after considering the Response, the Examiner would either allow the case or send another Office Action.

Double-Patenting Rejections

Claims 1 – 69 were provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 – 68 of co-pending Application No. 10/046,868, and claims 1 – 60 of co-pending Application No. 10/046,861. As noted in the previous Response, Applicant is willing to file Terminal Disclaimers if necessary to overcome these rejections in the event the conflicting claims are patented.

Section 103 Rejections

Claims 1 – 69 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application Publication 2001/0034881 (Washington). Applicant respectfully traverses this rejection.

Applicant reminds the Examiner that to establish prima facie obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. *In re Bond*, 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis.

In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings “must be clear and particular. ... Broad conclusory statements regarding the teaching of multiple references, standing alone, are not ‘evidence’.” *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

Applicant respectfully submits that there are numerous features and limitations of claim 1 that Washington fails to teach or suggest, and so Washington fails as a basis for establishing *prima facie* obviousness.

Claim 1 recites:

1. A method for executing graphical data flow programs, the method comprising:

executing a first graphical data flow program, wherein said executing the first graphical data flow program produces first program output;

executing a second graphical data flow program concurrently with the first graphical data flow program, wherein said executing the second graphical data flow program produces second program output; and

displaying the first program output and the second program output in a single graphical user interface on a display.

As noted in the previous Response, Washington nowhere teaches or suggests executing two graphical programs concurrently, where each graphical program produces respective program output. Nor does Washington teach or suggest, or even hint at, displaying the respective program outputs from the two concurrently executing graphical program in a single graphical user interface on a display. Nor does Washington provide any motivation to execute two graphical programs concurrently, nor to display the respective program outputs from the two concurrently executing graphical program in a single graphical user interface on a display.

In fact, Applicant further submits that displaying first program output from a first graphical data flow program and second program output from a second graphical data flow program in a single graphical user interface is a novel feature that is unknown in the prior art.

Again, Applicant submits that a single graphical user interface that displays program output from two different concurrently executing graphical data flow programs is a novel concept that is unknown in the prior art and *respectfully requests that the Examiner provide a reference that teaches the concept of such a single graphical user interface.*

Applicant notes that Washington is directed to enhancing the readability of a graphical program, (Abstract), and submits that Washington's disclosure of a graphical program generation (GPG) program is not germane to the present application. For example, Applicant notes that the present application makes no mention of a GPG program, nor or automatically generating graphical programs at all, although such generated programs are certainly not excluded from use in Applicant's system and method. It is unclear to Applicant why the Examiner has cited Washington's GPG program generation of graphical programs against the features and limitations of claim 1, since claim 1 (and the present invention in its entirety) makes no mention of this.

Also, Applicant notes that the Examiner has apparently combined two different examples of graphical program execution in Washington (0090, 0122, fig. 2; and 0110, 0162, 0165) in an attempt to construct (via hindsight analysis) the limitation of concurrently executing two graphical programs included in claim 1, which is improper.

The Examiner admits that "Washington's disclosure is not clear regarding the displaying one or more elements of the first and second graphical program in a same graphical user interface [sic]", but then goes on to assert various speculated properties of Washington's GPG program, including the unsupported assertion that "it is within the capability of the GPG to display one or more elements of the first and second graphical program in a same graphical user interface [sic]". Applicant notes that Washington nowhere discloses or indicates this alleged feature of Washington's GPG program.

The Examiner further asserts that "it would be naturally desirable to have a single GUI on which the user can control or monitor operations of the concurrently running multiple graphical programs", and then asserts "that it would have obvious to one of skill in the art, at the time the invention was made, to implement the execution of the second graphical program concurrently with the first one to measure operation performance of the being modeled physical device, and displaying of one or more elements of the first and second graphical program in a same graphical user interface, such as a front panel, to Washington, which implementation enables the user to control or monitor operations of the first and second graphical program. [sic]"

Applicant respectfully submits that the Examiner's arguments and conclusions are improper and incorrect, and submits that the Examiner has simply applied hindsight

analysis, using the Applicant's claims as a blueprint in an attempt to construct Applicant's invention as claimed, which is improper. In fact, Applicant respectfully submits that the Examiner has simply added Applicant's claimed features and limitations to Washington (and omitted primary aspects of Washington, such as the GPG program programmatically generating the graphical program(s)), in the attempt to construct Applicant's invention. Moreover, the only motivations suggested by the Examiner to modify Washington to include the presently claimed features and limitations is that "it would be naturally desirable", and that the implementation "enables the user to control or monitor operations of the first and second graphical program".

Applicant reminds the Examiner that per *In re Dembicza*k, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999), the art must fairly teach or suggest to one to make the specific combination as claimed. *That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.* Applicant respectfully submits that the Examiner has simply cited an improved result as motivation to modify Washington to include the features and limitations of claim 1, which is improper.

The Office Action also presented arguments involving Kodosky et al (USP 5,475,851, "Kodosky") and Microsoft Corporation's Multiple Documents Interface (MDI), although these additional references were not cited in the rejection, and no MDI reference was provided by the Examiner. For example, the Examiner asserted that Washington is an improvement over Kodosky, that Kodosky discloses concurrent execution of graphical programs, and that "combining Kodosky's teaching of concurrent running of data flow to Washington would have been obvious in light of Washington."

Applicant respectfully submits that neither Kodosky nor Washington provides a motivation to combine. For example, nowhere does Kodosky suggest the desirability of using a GPG program to programmatically generate a graphical program, nor, as noted above, does Washington suggest the desirability of executing multiple graphical programs concurrently. Thus, Applicant submits that the Examiner's attempted combination of Kodosky and Washington is improper.

The Office Action also asserts that “each node itself generate a data flow, and a diagram of connected nodes comprises multiple data flow (or subset of data flow) running concurrently”. Applicant respectfully submits that the Examiner has mischaracterized graphical program nodes.

For example, Applicant notes that a data flow diagram, also referred to as a block diagram of a graphical program, comprises *a plurality of interconnected nodes that visually represent functionality of the data flow diagram*. Thus, *a single node is not properly considered to be a data flow diagram or graphical program*, and so Applicant submits that a single diagram of connected nodes is *not* the same as multiple data flow diagrams running concurrently.

Moreover, regarding the Examiners assertion that the interconnected nodes in a single data flow diagram execute concurrently, Applicant submits that single nodes in a graphical program are not themselves graphical programs or data flow diagrams, they do not teach this feature of the claimed invention.

Regarding the Examiner’s citation of MDI (without reference), Applicant respectfully submits that MDI (Multiple Document Interface) is non-analogous art and is particularly not germane to graphical programming, nor to the display in one GUI of program output from each of two concurrently executing graphical programs.

As is well known, MDI is a Microsoft Windows API (Application Programming Interface) that enables programmers to create applications with multiple windows, where each MDI application has a single *main window*, and any number of *child windows*, and where the child windows are displayed within the main window.

Applicant further submits that MDI is generally used to allow an application, e.g., Excel, Word, etc., to have a single parent window and multiple child windows that are constrained to appear within the parent window.

Applicant submits that MDI has nothing to do with graphical programs, nor graphical programming, and more specifically does not address or support displaying respective output from two concurrently executing graphical programs in a single GUI. Applicant respectfully submits that even were the Examiner to combine Washington, Kodosky, and MDI, the resulting combination would still not produce Applicant’s

invention as claimed. It is unclear to Applicant how one would apply MDI to the domain of graphical programming, and specifically, to the display of program output from multiple concurrently executing graphical programs, nor does the Examiner explain how this functionality could be accomplished.

Thus, for at least the reasons provided above, Applicant respectfully submits that claim 1, and claims dependent thereon, are patentable over Washington. Independent claims 28, 36, 38, 40, and 51-54 recite similar features as claim 1, and so for at least the reasons provided above, Applicant submits that these claims, and claims respectively dependent thereon, are also allowable.

Regarding claims 55, 62, and 66, Applicant submits that Washington also fails to teach or suggest all the features and limitations of these claims.

Claim 55 recites:

55. A method for performing a software simulation, the method comprising:
executing a simulation program, wherein the simulation program comprises a first graphical program;
executing a measurement program concurrently with the simulation program, wherein the measurement program comprises a second graphical program; and
displaying a single graphical user interface comprising a first plurality of graphical user interface elements for the simulation program and a second plurality of graphical user interface elements for the measurement program.

Applicant respectfully submits that Washington nowhere teaches or suggests concurrent execution of a simulation program comprising a first graphical program and a measurement program comprising a second graphical program, nor displaying a single graphical user interface comprising a first plurality of graphical user interface elements for the simulation program and a second plurality of graphical user interface elements for the measurement program. Nor does Washington provide any suggestion that would motivate one to modify Washington to include these features and limitations.

Thus, for at least the reasons provided above, Applicant respectfully submits that claim 55, and claims dependent thereon, are patentable over Washington. Independent

claims 62, and 66 recite similar features as claim 55, and so for at least the reasons provided above, Applicant submits that these claims, and claims respectively dependent thereon, are also allowable.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over Washington. For example, claim 2 recites the additional limitations of:

receiving user input to the single graphical user interface during said executing; and
providing the user input to at least one of the first graphical data flow program or the second graphical data flow program.

Applicant submits that the concept of receiving user input to a single graphical user interface that displays program output from two different concurrently executing graphical data flow programs and providing the user input to at least one of the concurrently executing graphical data flow programs is a novel concept that is unknown in the prior art. Washington contains no teaching or suggestion that would motivate one to implement this functionality, for reasons similar to those discussed above.

As another example, claim 5 recites the additional limitations of:

wherein the first graphical data flow program executes on a first computer system;
wherein the second graphical data flow program executes on a second computer system.

Washington does not teach or suggest the concept of executing a first graphical data flow program on a first computer system and concurrently executing a second graphical data flow program on a second computer system.

As another example, claim 18 recites the additional limitations of:

wherein the first graphical data flow program is associated with a first graphical program development environment;
wherein the second graphical data flow program is associated with a second graphical program development environment, wherein the second graphical program development environment is different than the first graphical program development environment.

Washington contains no teaching or suggestion that would motivate one to perform a concurrent execution of two graphical data flow programs that are created in two different graphical program development environments. Furthermore, the concept of a single graphical user interface that displays program output from two concurrently executing graphical data flow programs created in different graphical program development environments is a novel concept that is unknown in the prior art, and Washington contains no teaching or suggestion that would motivate one to veer from the prior art in this regard.

Applicant thus submits that the present claims are patentable over Washington for at least the reasons given above.

Removal of the 103 rejection of claims 1-69 is respectfully requested.

CONCLUSION

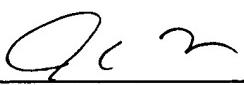
In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-56000/JCH.

Also enclosed herewith are the following items:

- Return Receipt Postcard
- Petition for Extension of Time
- Check in the amount of \$ for fees ().
- Other:

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 6/28/2005 JCH/MSW